УДК 004.633
С. И. Пухов, В. Ю. Носков
ФГАОУ ВО «Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина», г. Екатеринбург, Россия

РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ОБНОВЛЕНИЯ СИСТЕМНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В РАСПРЕДЕЛЕННЫХ СЕТЯХ АПК

Аннотация

Данная работа посвящена разработке метода автоматизированного обновления системного программного обеспечения в распределенных сетях аппаратно-программного комплекса и несет в себе практическую значимость в задачах разработки. Во введении обосновывается актуальность выбранной темы, формулируются цель и задачи исследования, указывается объект и предмет исследования.

Рассматриваются такие задачи, как поиск оптимального формата хранения данных и протокола передачи данных, написание скриптов на языке Python, тестирование созданной системы. Поиск оптимального формата хранения данных и протокола передачи являются первоочередными задачами, так как при большом количестве аппаратно-программных комплексов объем данных и скорость передачи будут играть немаловажную роль в автоматическом процессе обновления системного ПО. Созданная система поможет оптимизировать и автоматизировать обновление программного обеспечения в аппаратно-программных комплексах.

Ключевые слова: *автоматизированное обновление, АПК, система контроля версий, программное обеспечение, репозиторий, python, Linux.*

Abstract

This work is devoted to the development of a method for automated updating of system software in distributed networks of a hardware-software complex and carries with it practical significance in development tasks. In the introduction, the relevance of the chosen topic is substantiated, the goal and objectives of the research are formulated, the object and subject of research are indicated.

Tasks such as finding the optimal data storage format and data transfer protocol, writing scripts in Python, testing the created system are considered. Finding the best format for storing data and the transfer protocol is a top priority, since with a large number of hardware and software systems, the amount of data and the transfer rate will play an important role in the automatic process of updating system software. The created system will help to optimize and automate software updates in hardware and software complexes.

Key words: *automated update, hardware-software complex, version control system, software, repository, python, Linux.*

Введение. Любой персональный компьютер, даже самый современный, без установленного на него необходимого ПО является просто грудой железа. В таком состоянии ни один компьютер не способен выполнить даже самые элементарные операции. Именно поэтому существует огромное множество различного ПО. Каждый пользователь устанавливает определенный набор программ на свое оборудование для выполнения конкретных задач.

У каждой разрабатываемой программы существует жизненный цикл. За время жизни ПО чаще всего возникают изменения в его работе. Изменения

бывают разными — от исправления ошибки, добавления или изменения функционала до полного переписывания. В большинстве случаев название программы остаётся тем же, изменяется подназвание — так называемая версия [1]. Версия программы может быть целым числом (Corel Draw 11), последовательностью чисел (JDK 1.0.3), годом (Windows 2000) или текстом (Embarcadero Delphi XE).

В системах Linux автоматизированное обновление системного ПО зависит от конкретного дистрибутива GNU/Linux. Для каждого дистрибутива существует свой набор команд и серверов по автоматизированному обновлению системного ПО.

Именно из-за того, что для каждой системы выпускается свой набор программ и их версий, было принято решение сделать свой универсальный метод для автоматизированного обновления системного ПО.

Информационная система предназначена для автоматического обновления системного ПО в ОС Linux и упрощения работы пользователя.

Главная цель создания информационной системы – обновление ПО до стабильной свежей версии без участия пользователя и ошибок в дальнейшей работе.

Для достижения поставленной цели потребовалось решить следующие задачи:

1) определиться с форматом хранения данных о системном ПО;
2) определится с протоколом передачи данных;
3) написать серверную часть на Python 2.7.5;
4) написать клиентскую часть на Python 2.7.5;
5) автоматизировать клиентскую и серверную части;

Описание разработки. Для создания системы требуются знания языка программирования Python и навыки работы с Linux OS.

Процесс разработки. Для автоматизированного обновления системного ПО был использован метод централизованной системы управления версиями. Ведущим устройством является сервер, ведомым – клиент. Сервер в системе один, а клиентов несколько.

Для хранения данных был выбран формат xml. Этот формат очень удобен для хранения данных. Документы в формате XML содержат данные, заключенные в теги [2]. Теги определяют структуру и смысл данных — то, чем они являются. Также можно использовать атрибуты для тегов, например, для написания нужных нам параметров.

```
<?xml version="1.0"?>
<versions_server>
    <virtual_machines>
        <machine1>
            <machine1-tmp last_update="2019-02-02 16.48.15" hash_sum="AB3CAFF61455D3F6D56BEEA7F4AC369A34828F3B" version="1.0"></machine1-tmp>
            <machine1-tmp last_update="2019-03-05 15.35.44" hash_sum="FC8B78123B7E619335B80F0BE26271A575105FD1" version="1.1"></machine1-tmp>
        </machine1>
        <machine2>
```

Рис. 1. Примерный вид хранения данных в xml

На сервере хранится эталонный файл xml, в котором указываются все наименования ПО. А для каждого отдельного ПО указывается дата выпуска обновления, хеш-сумма и номер версии.

Для передачи данных между сервером и клиентами был выбран протокол SSH.

SSH – сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений для передачи файлов [3]. При передаче шифрует весь трафик, включая и передаваемые пароли. SSH допускает выбор различных алгоритмов шифрования. SSH-клиенты и SSH-серверы доступны для большинства сетевых операционных систем.

После выбора формата хранения данных и протокола передачи информации пришло время написать скрипты для серверной части на языке Python. Для работы с форматом xml потребовалась библиотека xml.etree.ElementTree. Чтобы работать с файловой структурой и Linux OS потребовались библиотеки subprocess и os. А для передачи данных через ssh была использована библиотека pexpect. Для формирования логирования библиотеки – logging и time.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import subprocess
import time
import xml.etree.ElementTree as Etree
import pexpect
import os
import logging
```

Рис. 2. Фрагмент скрипта по заполнению xml файла на клиенте

Серверная часть состоит из нескольких скриптов:

1. Скрипт по начальной настройке клиентов.
2. Скрипт по рассылке файлов для клиентов.
3. Скрипт по формированию эталонного xml файла.
4. Скрипт по сравнению хеш-сумм на клиенте и сервере.
5. Скрипт по работе с виртуальными машинами.
6. Скрипт по управлению клиентами.
7. Скрипт по сравнению версий ПО между сервером и клиентами.

Реализация клиентской части была проще серверной. На клиенте лишь выполнялись команды, которые передавал сервер.

Клиентская часть состояла из 3 скриптов:

1. Скрипт по проверке нужных файлов и каталогов.
2. Скрипт по подсчету хеш-суммы на клиенте и формирования данных в файл xml.
3. Скрипт по обновлению ПО на клиенте.

На рисунке 3 представлена часть кода по подсчету хеш-суммы ПО на клиенте.

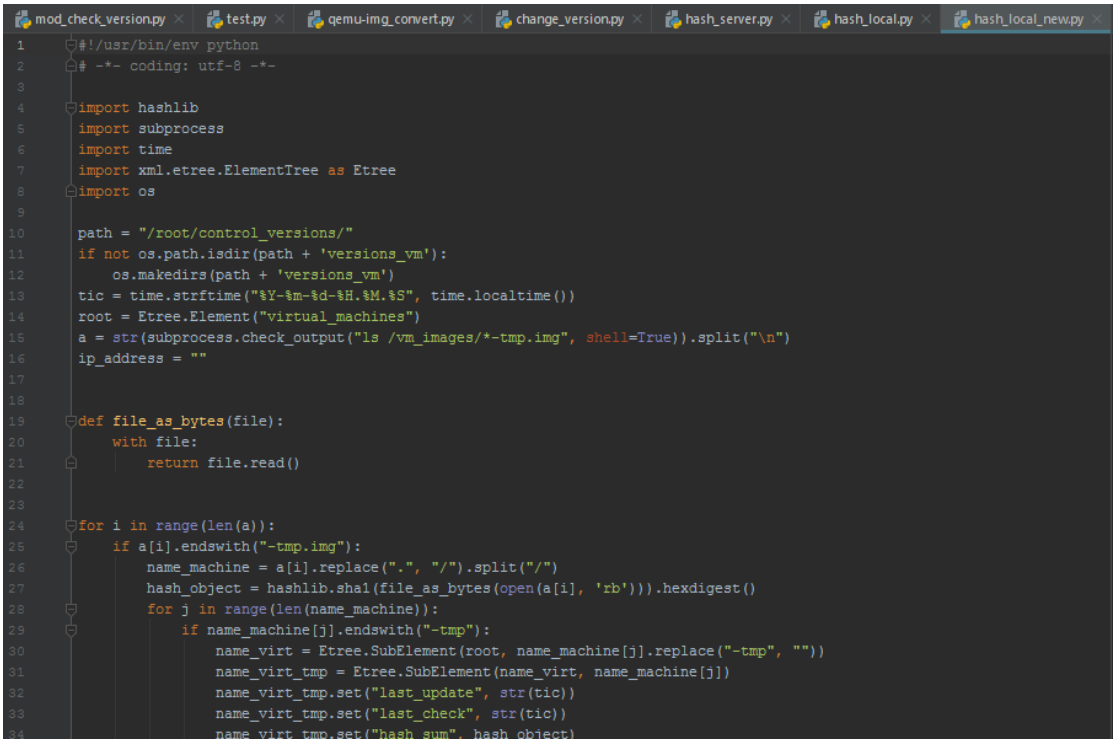

Рис. 3. Фрагмент скрипта по заполнению xml файла на клиенте

Как для серверной, так и для клиентской частей были написаны несколько скриптов. Каждый выполнял свои функции. Скрипты в серверной части были добавлены в файл cron в Linux OS для запуска по времени.

Таким образом была разработана система по автоматизированному обновлению ПО в распределенных сетях АПК.

Заключение. Созданная информационная система должны облегчить обновление системного ПО под ОС Linux. Также система должна снизить влияние человеческого фактора на установку программ и именно поэтому снизить количество ошибок в дальнейшей работе.

Список использованных источников

1. Нумерация версий программного обеспечения. [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Нумерация_версий_программного_обеспечения.

2. XML для начинающих. [Электронный ресурс]. – Режим доступа: https://support.office.com/ru-ru/article/xml-%D0%B4%D0%BB%D1%8F-%D0%BD%D0%B0%D1%87%D0%B8%D0%BD%D0%B0%D1%8E%D1%89%D0%B8%D1%85-a87d234d-4c2e-4409-9cbc-45e4eb857d44.

3. SSH. [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/SSH.